

Arquitecturas para tratar grandes volúmenes de información

Procesamiento de Datos a Gran Escala

Cluster Hadoop

- Introducción a Hadoop
- Hadoop se puede instalar de tres maneras distintas:
 - Standalone
 - Pseudo-Distributed
 - Fully Distributed

Google Origins

The Google File System

2003 Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung
Google*



MapReduce: Simplified Data Processing on Large Clusters

2004 Jeffrey Dean and Sanjay Ghemawat
jeff@google.com, sanjay@google.com
Google, Inc.



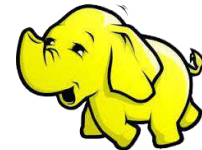
Bigtable: A Distributed Storage System for Structured Data

2006 Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach
Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber
(fay.jeff.sanjay.wilson.chandra.tushar.fikes.gruber}@google.com
Google, Inc.



Abstract
Bigtable is a distributed storage system for managing structured data that is designed to scale to a very large number of nodes and to support a very large number of users. Many projects at Google store data in Bigtable, including web indexing, Google Earth, and Google Finance. These applications place very different demands on Bigtable, both in terms of data size (from URLs to

achieved scalability and high performance, but Bigtable provides a different interface than such systems. Bigtable does not support a full relational data model; instead, it provides clients with a simple data model that supports dynamic control over data layout and format, and allows clients to reason about the locality properties of data represented in the underlying storage. Data is indexed using row and column names that can be arbitrary strings. Bigtable also treats data as uninterpreted strings.



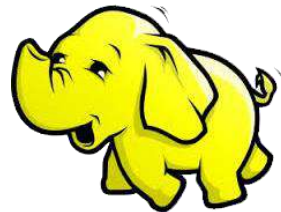
What is Hadoop?

- **Hadoop:**

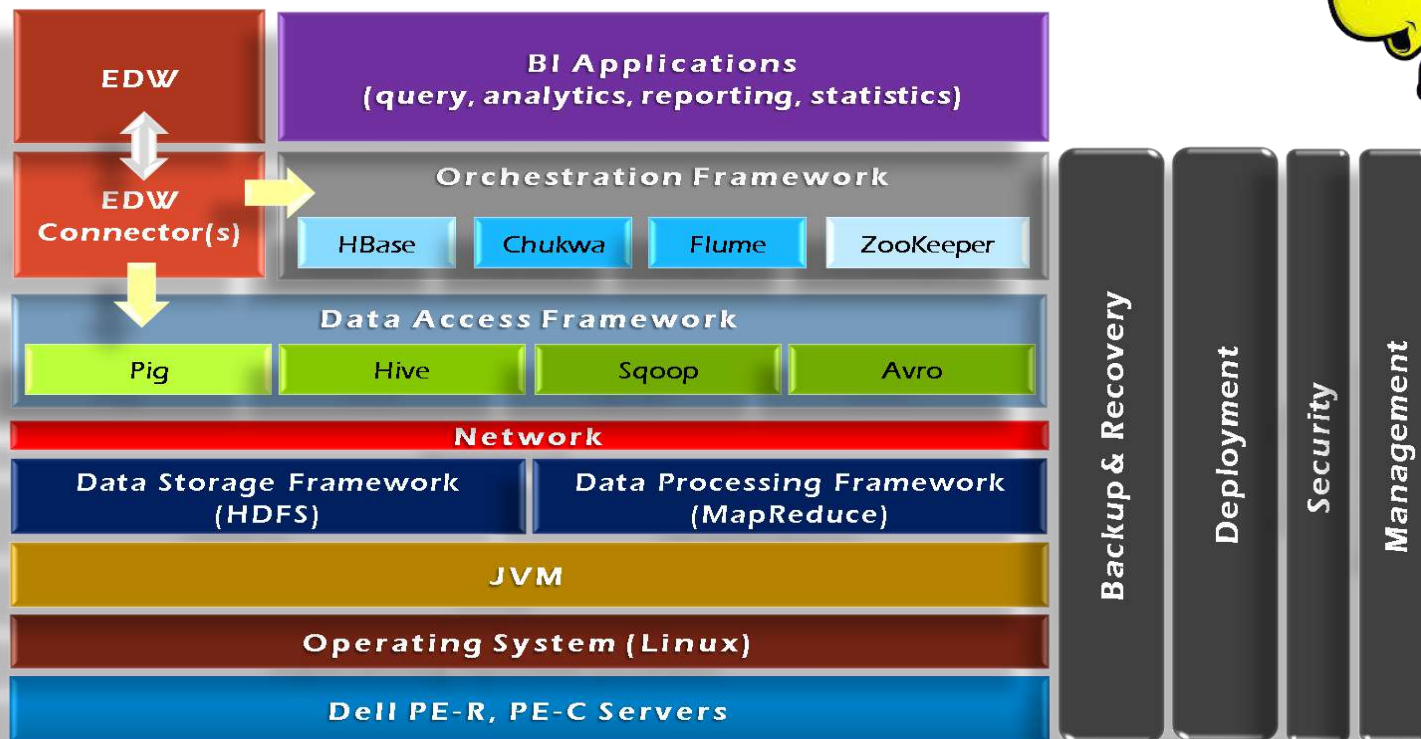
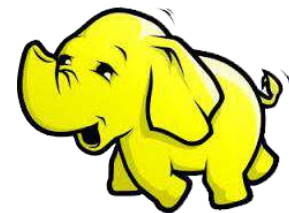
- An open-source software framework that supports data-intensive distributed applications, licensed under the Apache v2 license.

- **Goals / Requirements:**

- Abstract and facilitate the storage and processing of large and/or rapidly growing data sets
 - Structured and non-structured data
 - Simple programming models
- High scalability and availability
- Use commodity (cheap!) hardware with little redundancy
- Fault-tolerance
- Move computation rather than data

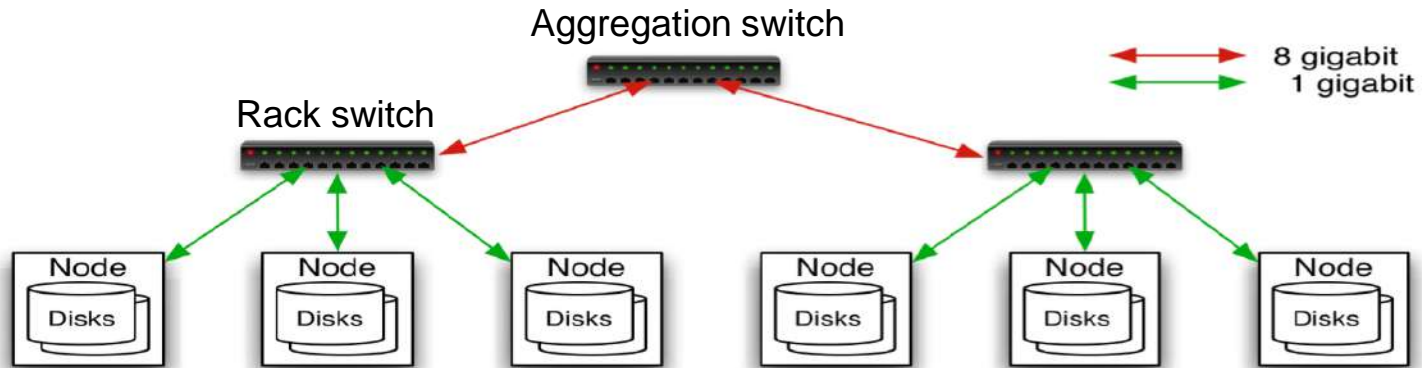
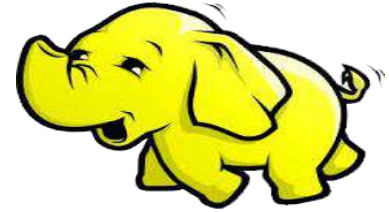


Hadoop Framework Tools

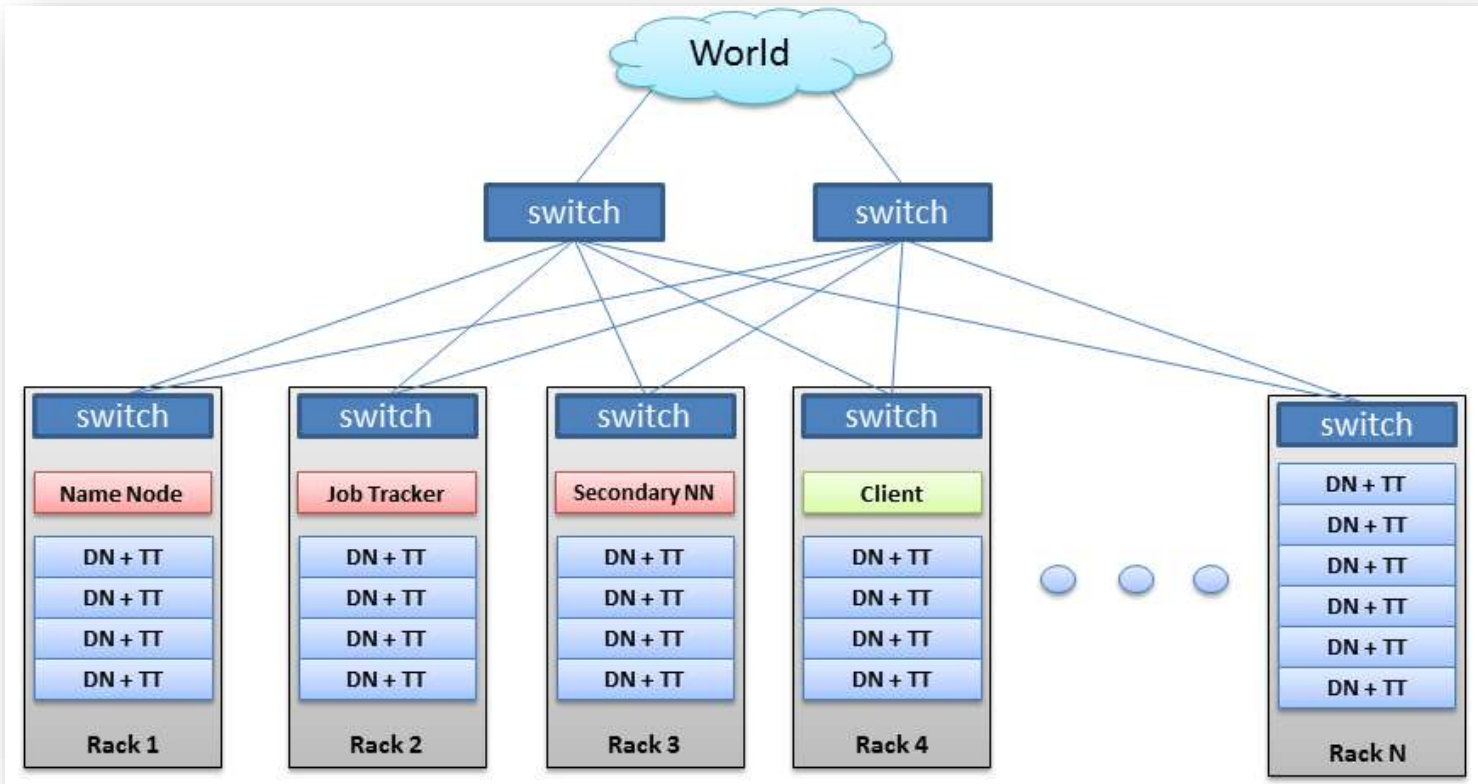


Cluster Hadoop

- Cluster creado con commodity Hardware;
 - Nodos inicialmente eran PCs
 - 30-40 nodos/rack
 - Red a 1 gigabit/s en rack



Arquitectura de almacenamiento HDFS

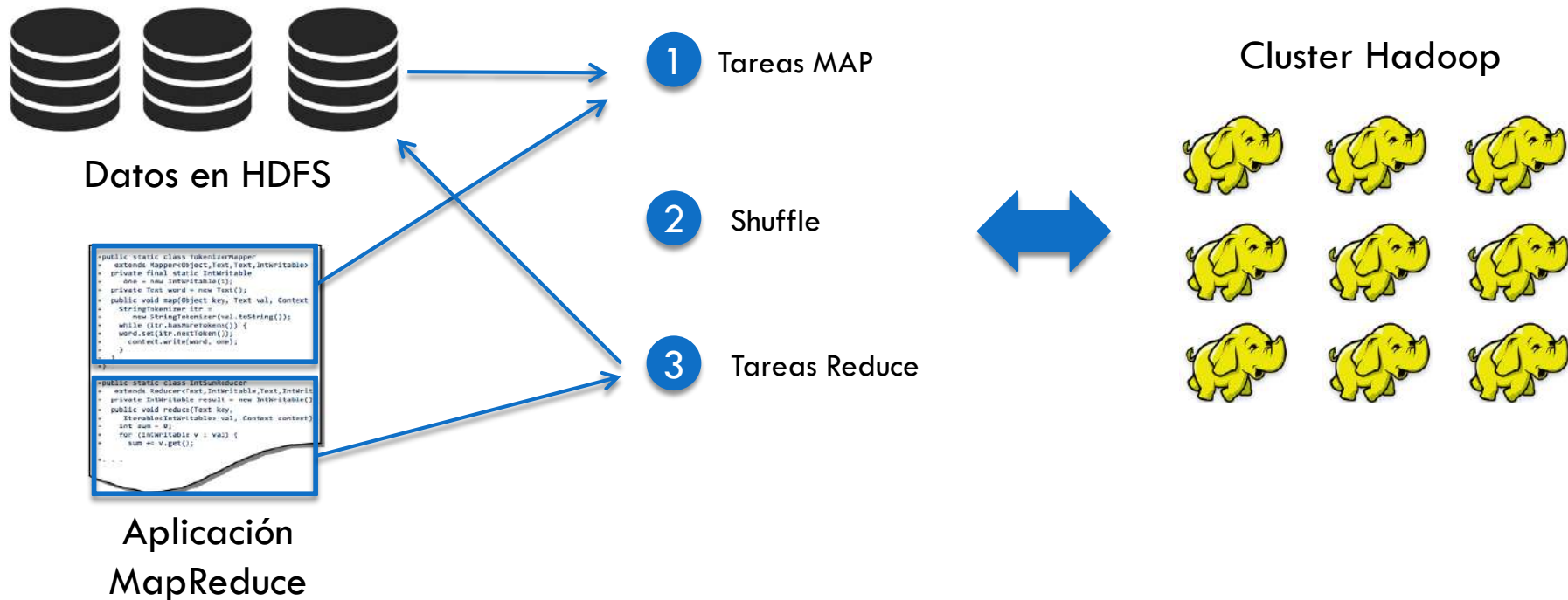


HDFS : Hadoop Distributed File System



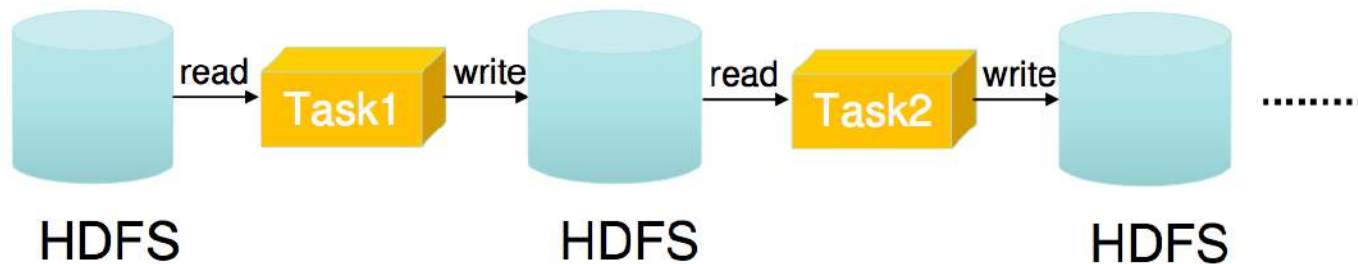
- Sistema de Ficheros distribuido muy grande
 - 10K nodos, 100 millones de ficheros 10PB
- Realizado con “*Commodity Hardware*”
 - Ficheros replicados para tolerancia a fallos
 - Detecta fallos y recupera los datos.
- Optimizado para proceso por lotes (“*Batch Processing*”).
 - Expone la localización de los datos y así permite que la computación se pueda llevar cerca de los datos.
 - El ancho de banda agregado es muy alto.

Cómo se ejecuta una aplicación Hadoop

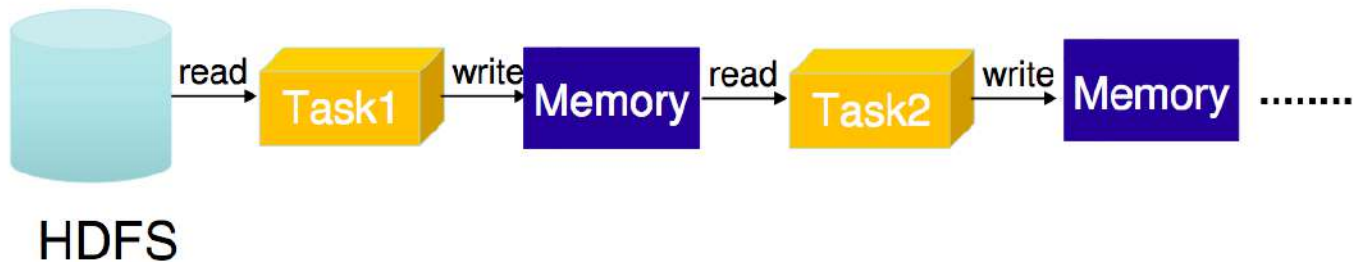


Intro: MapReduce vs Spark

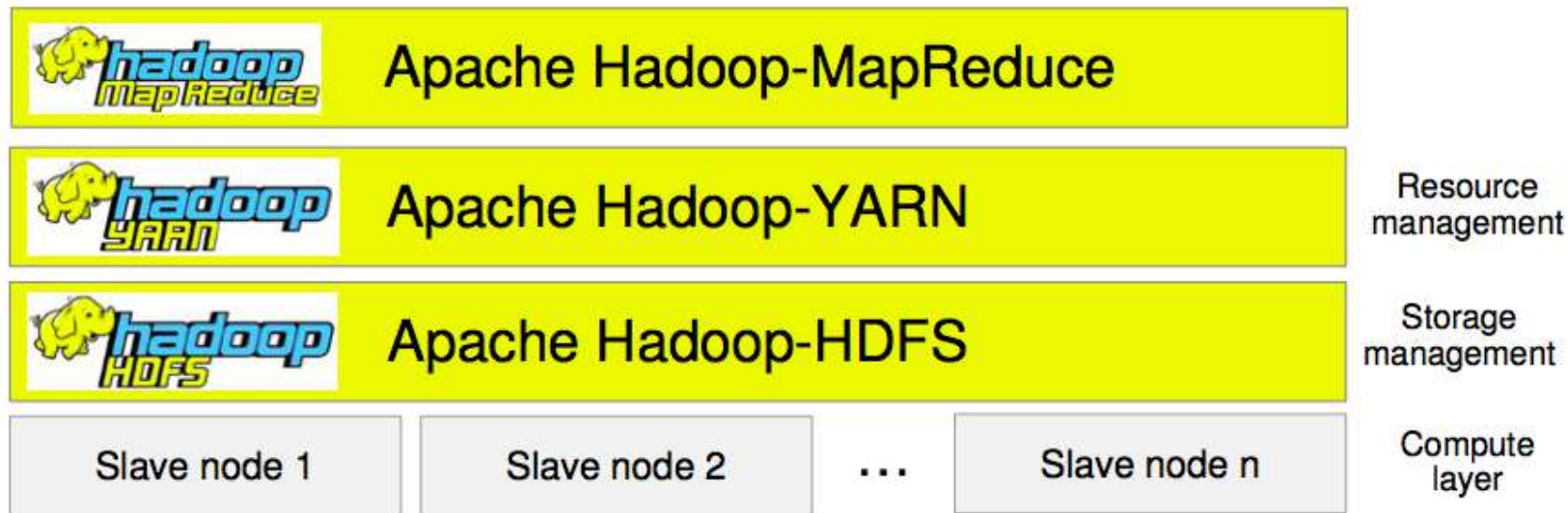
- Map/Reduce:



- Spark:



Intro: Arquitectura Hadoop



¿Qué necesitamos?

- Una ordenador con software para crear máquinas virtuales y conexión a Internet
 - La instalación en una máquina real es equivalente
 - Usaremos VMware
- Una ISO de Linux
 - Usaremos CentOS que es una RedHat “opensource”
- Hadoop
 - Este y otro software relacionado lo descargaremos una vez instalado Linux

Instalación de Hadoop

- Partimos de una imagen de MV que encontraréis en cada equipo
 - Recomendación: cambiar la configuración de la MV para “darle potencia”, ya que los equipos del laboratorio nos lo permiten
 - 2 GB de RAM
 - 2 procesadores
 - Acceso:
 - `bigdata (bigdata)`
 - `root (bigdata)`



Antes de empezar, copiaros la imagen de MV tal cómo la encontréis en vuestro equipo, pues la utilizaremos varias veces como punto de partida

Máquina virtual de partida

- Instalación “minimalista”
 - CentOS 7 (ISO disponible en los equipos del laboratorio)
 - 2 GB de RAM
 - 1 procesador
 - Instalación paquete de software “Compute node”
 - Se añade:
 - Herramientas de monitorización de HW
 - Herramientas de rendimiento
 - Administración remota de Linux
 - Herramientas de desarrollo
 - Habilitar interfaz de red

Máquina virtual de partida

- Actualizar repositorios
 - yum update
- Paquetes adicionales a instalar
 - yum install <paquete>
 - xfsprogs
 - openssh

Máquina virtual de partida

- Entorno gráfico
 - `yum install <paquete>`
 - `gnome-classic-session`
 - `gnome-terminal`
 - `gnome-terminal-nautilus`
 - `control-center`
 - `liberation-mono-fonts liberation-sans-fonts`
 - `Firefox`
 - `yum groupinstall <grupo-paquetes>`
 - “GNOME Desktop”
 - “Graphical Administration Tools”
- Habilitamos arranque en modo gráfico
 - `systemctl set-default graphical.target`

Instalar Linux

- **Actualizar sistema**

- `> yum -y update`

- **Descargar wget (herramienta de descarga web)**

- `> yum -y install wget`

- **Crear usuario 'estudiante'**

- La idea es trabajar con este usuario y no como root

- `> useradd bigdata`

- `> passwd bigdata` (introduce la contraseña)

Instalar Hadoop

- La instalación que vamos a realizar es Single Node (una sola máquina)
- Previamente necesitamos instalar:
 - Java 7. La opción más sencilla en OpenJDK que viene con CentOS
 - `> yum -y install java-1.7.0-openjdk`
 - SSH. Viene con Linux, pero necesitamos rsync
 - `> yum -y install ssh rsync`
- Y ahora hadoop...
 - `> wget apache.rediris.es/hadoop/common/hadoop-2.8.1/hadoop-2.8.1.tar.gz`

Instalar Hadoop

- **Descomprimir hadoop**
 - `> tar xvzf hadoop-2.8.1.tar.gz`
- **Movemos a /opt**
 - `> mv hadoop-2.8.1 /opt`
- **Creamos un link a la carpeta de hadoop**
 - Permite elegir entre diferentes versiones
 - `> cd /opt`
 - `> ln -s hadoop-2.8.1 hadoop`
 - `> cd hadoop`
- **Editar fichero `etc/hadoop/hadoop-env.sh`**
 - `export JAVA_HOME=/usr/lib/jvm/jre-1.7.0-openjdk`

Instalar Hadoop

- Hemos descomprimido el código de hadoop dentro de nuestro sistema de ficheros local
- ¿Por qué no hay que compilar nada?
 - Hadoop es código JAVA!
- Conviene chequear compatibilidad con tu versión de Java
 - <http://hadoop.apache.org/releases.html>
 - <https://wiki.apache.org/hadoop/HadoopJavaVersions>

EL lenguaje de programación Java

➤ Máquina Virtual de Java (JVM)

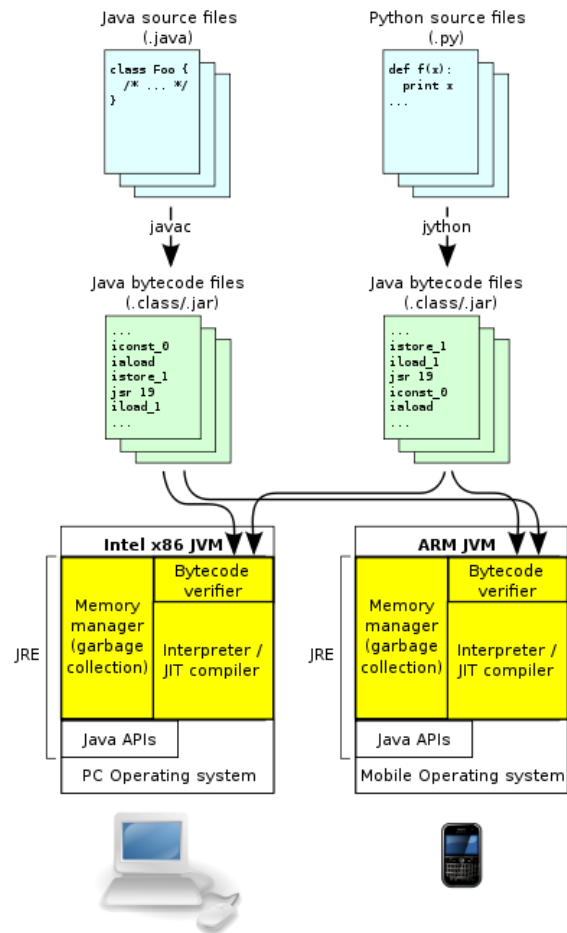
➤ Portabilidad

- Cualquier dispositivo
- Cualquier SO
- Java > ByteCode > Código ejecutable

➤ Menor rendimiento

- Etapas intermedias de ejecución
- Dificultad de programación MP/MC

➤ Aislamiento

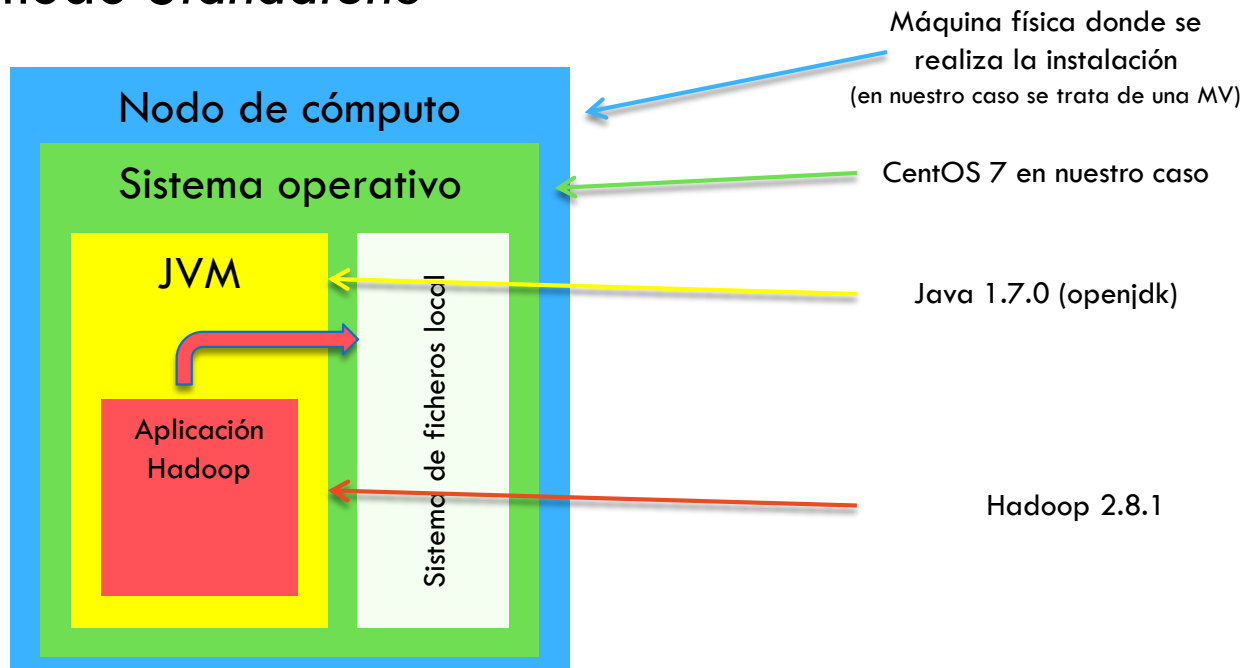


Instalación de Hadoop

- Instalación en modo *Standalone*
 - No se ejecutan demonios
 - Todo se ejecuta en una única Máquina Virtual de Java (MVJ)
 - No se usa HDFS
 - Adecuado para desarrollo y debug de aplicaciones

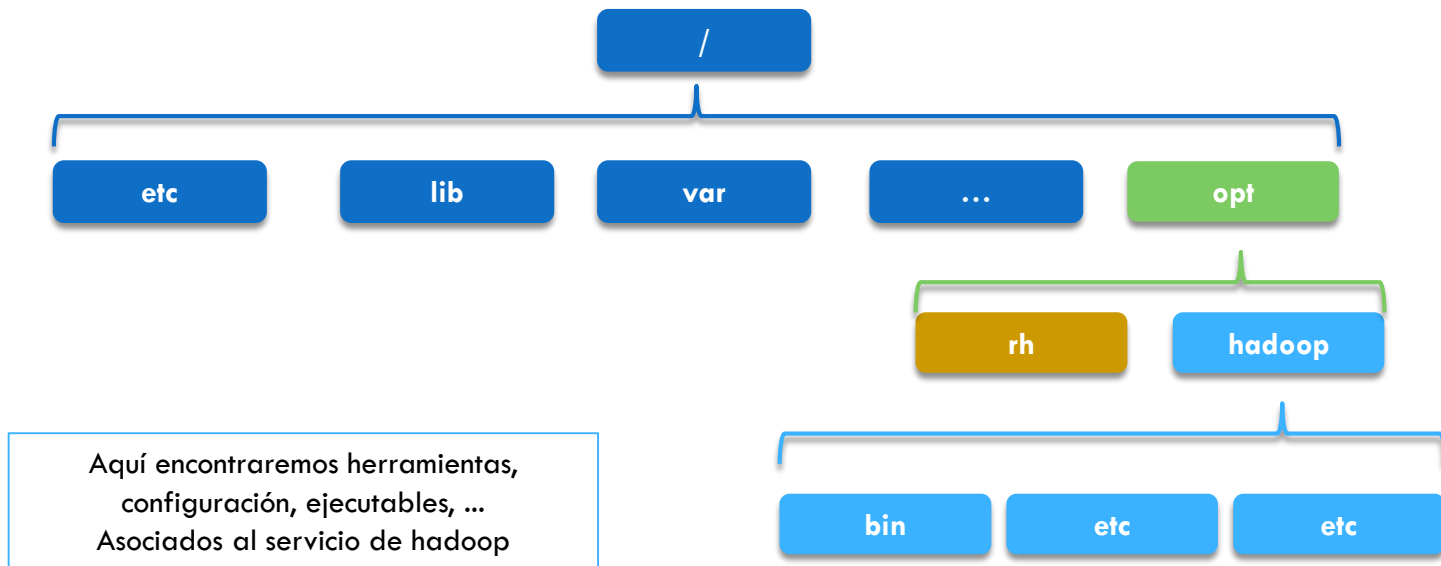
Instalación de Hadoop

➤ Instalación en modo *Standalone*



Instalación de Hadoop

➤ Instalación en modo *Standalone*



Instalar Hadoop - Standalone

- Por defecto Hadoop está configurado para ejecutarse en modo non-distributed, como un único proceso java
- Ejemplo 1
 - `> cd /opt/hadoop`
 - `> bin/hadoop`
 - `> mkdir prueba`
 - `> cp etc/hadoop/*.xml prueba`
 - `> bin/hadoop jar share/hadoop/tools/lib/hadoop-streaming-2.8.1.jar -input prueba -output salida -mapper cat -reducer wc`



No se usa
HDFS!!

Instalar Hadoop - Standalone

➤ Ejemplo 1

➤ > bin/hadoop

jar

Clase a invocar

share/hadoop/tools/lib/hadoop-streaming-2.8.1.jar

-input prueba

Ficheros (en este caso directorio)

-output salida

Directorio de salida

-mapper cat

Qué hacer en las fase map y reduce

-reducer wc

Parámetros de la clase

Instalar Hadoop - Standalone

➤ Ejemplo 1

➤ Comprobar la salida

➤ `> cat salida/part-00000`

➤ Comparar resultado con el siguiente comando

➤ `> wc prueba/*`

Instalar Hadoop - Standalone

➤ Ejemplo1 - Observaciones

➤ NO hemos utilizado HDFS en ningún momento

➤ Sólo se ha utilizado el FS local

➤ Directorios “prueba” y “salida”

➤ Repetir la ejecución de la prueba mientras en otra terminal ejecutas el comando “top”

```
done. And is in the process of committing
7/09/13 04:31:50 INFO mapred.LocalJobRunner: Records R/W=173/1
7/09/13 04:31:50 INFO mapred.Task: Task 'attempt_local121213557_0001_m_000001_0'
done.
7/09/13 04:31:50 INFO mapred.LocalJobRunner: Finishing task: attempt_local12121355
7/09/13 04:31:50 INFO mapred.LocalJobRunner: Starting task: attempt_local121213557
7/09/13 04:31:50 INFO output.FileOutputCommitter: File Output Committer Algorithm
version is 1
7/09/13 04:31:50 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup
temporary folders under output directory:false, ignore cleanup failures: false
7/09/13 04:31:50 INFO mapred.Task: Using ResourceCalculatorProcessTree: [ ]
7/09/13 04:31:50 INFO mapred.MapTask: Processing split: file:/opt/hadoop-2.8.1/pru
ba/capacity-scheduler.xml:0+4942
7/09/13 04:31:50 INFO mapred.MapTask: numReduceTasks: 1
7/09/13 04:31:50 INFO mapreduce.Job: map 100% reduce 0%
7/09/13 04:31:50 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(164857584)
7/09/13 04:31:50 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
7/09/13 04:31:50 INFO mapred.MapTask: soft limit at 83886080
7/09/13 04:31:50 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857500
7/09/13 04:31:50 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
7/09/13 04:31:50 INFO mapred.MapTask: Map output collector class = org.apache.hado
```

```
top - 04:31:48 up 9:29, 3 users, load average: 0,74, 0,27, 0,17
Tasks: 177 total, 1 running, 176 sleeping, 0 stopped, 0 zombie
%Cpu(s): 85,6 us, 14,0 sy, 0,0 ni, 0,0 id, 0,3 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem : 999920 total, 79424 free, 661000 used, 259496 buff/cache
KiB Swap: 2097148 total, 2037544 free, 59604 used, 118044 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
30004	bigdata	20	0	2244584	136928	20396	S	61,8	13,7	0:04.27	java
12124	bigdata	20	0	1492928	174780	12356	S	17,3	17,5	2:26.88	gnome-shell
1134	root	20	0	256144	42440	2324	S	8,3	4,2	1:03.66	Xorg
13670	bigdata	20	0	555324	14952	5208	S	1,7	1,5	0:21.34	gnome-term
25	root	20	0	0	0	0	S	0,7	0,6	0:10.65	kswapd0
709	root	20	0	302772	1836	1308	S	0,3	0,2	0:43.74	vimtoolsd
11883	bigdata	20	0	35988	1876	560	S	0,3	0,2	0:09.90	dbus-daemon
12204	bigdata	20	0	575276	5848	1564	S	0,3	0,6	0:04.83	caribou
29468	root	20	0	157704	2300	1568	R	0,3	0,2	0:03.77	top
1	root	20	0	128088	4252	2452	S	0,6	0,4	0:09.65	systemd
2	root	20	0	0	0	0	S	0,6	0,6	0:01.79	kthreadd
3	root	20	0	0	0	0	S	0,6	0,6	0:01.81	ksoftirqd/0
7	root	rt	0	0	0	0	S	0,6	0,6	0:00.00	migration/0
8	root	20	0	0	0	0	S	0,6	0,6	0:00.00	rcu_bh
9	root	20	0	0	0	0	S	0,6	0,6	0:03.30	rcu_sched
10	root	rt	0	0	0	0	S	0,6	0,6	0:16.53	watchdog/0

Instalar Hadoop - Standalone

➤ Ejemplo2

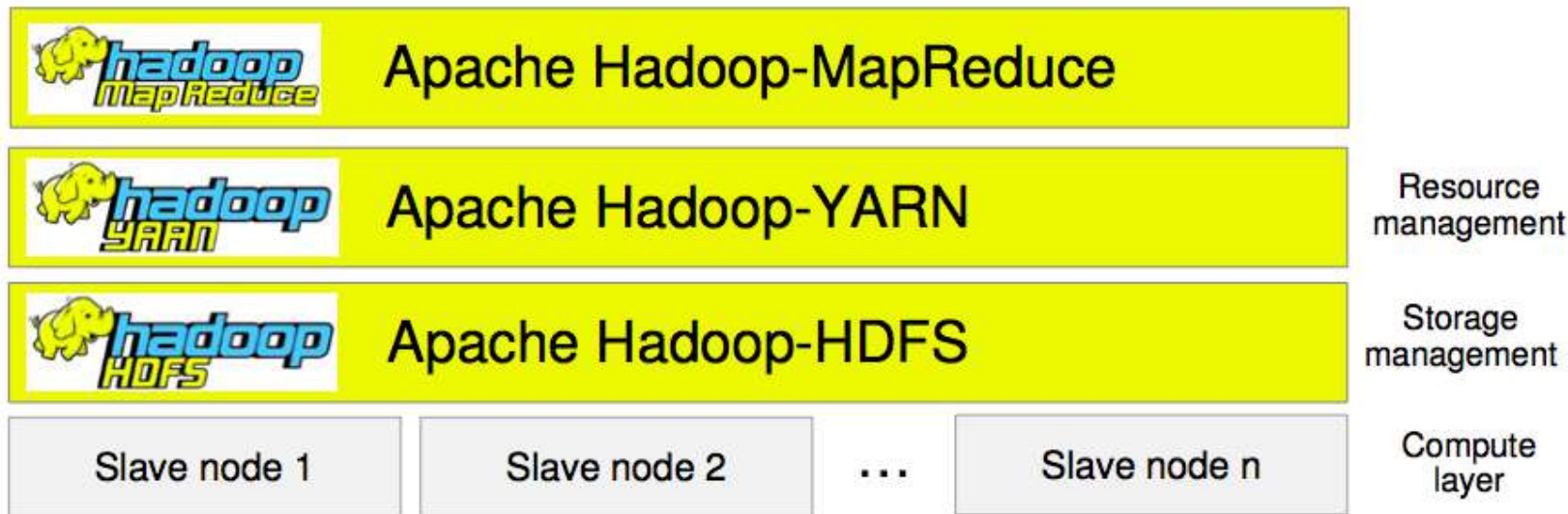
- Ejecutamos un wordcount con fichero de texto que creemos

- Pista:

- `share/hadoop/mapreduce/hadoop-mapreduce-examples-2.8.1.jar`

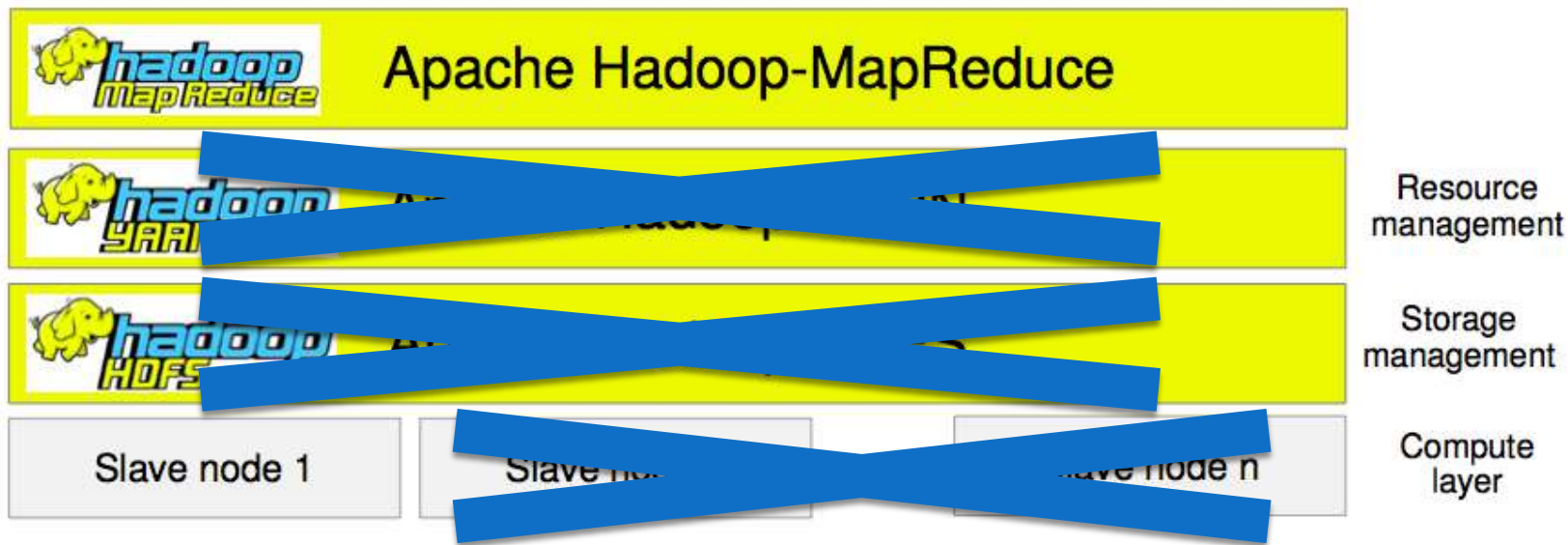
Instalación de Hadoop

- Este no el modo de ejecución de Hadoop del que estamos acostumbrados a hablar



Instalación de Hadoop

➤ Modo *standalone*



Instalación de Hadoop

- Hadoop se puede instalar de tres maneras distintas:
 - Standalone
 - No se ejecutan demonios
 - Todo se ejecuta en una única Máquina Virtual de Java (MVJ)
 - No se usa HDFS
 - Adecuado para desarrollo y debug de aplicaciones

Instalación de Hadoop

- Hadoop se puede instalar de tres maneras distintas:
 - Pseudo-Distributed
 - Todos los demonios se ejecutan en la misma máquina
 - En su propia MVJ
 - Se emplea HDFS
 - Adecuado para simular un cluster en una sola máquina y para debug de programas antes de llevarlos a un “cluster real”


Instalar Hadoop – Pseudo-Distributed

- Antes de nada debemos configurar ssh para funcionar sin contraseña para conexiones a la misma máquina (localhost)
 - En esta configuración los diferentes procesos de Hadoop hacen uso de la red para conectarse aunque estén en la misma máquina
 - `> ssh-keygen` (pulso Intro a todo)
 - `> ssh-copy-id localhost` (esto vale para cualquier máquina)
- Comprobar que se puede conectar
 - `> ssh localhost`
- Se pueden repetir los dos últimos pasos para 'hadoop-master', aunque no es necesario

Instalar Hadoop – Pseudo-Distributed

- Añade la siguiente configuración a *etc/hadoop/core-site.xml*:

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>
```



Importante: Se parte del directorio donde esta instalado Hadoop

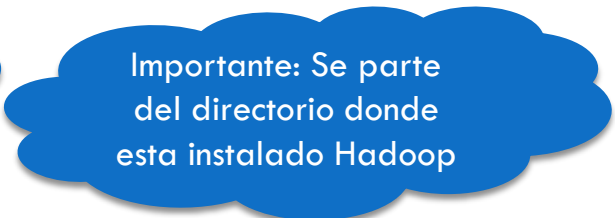
- Añade la siguiente configuración a *etc/hadoop/hdfs-site.xml*:

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
</configuration>
```

Instalar Hadoop – Pseudo-Distributed

- Formatear sistema de ficheros

- `> bin/hdfs namenode -format`



Importante: Se parte del directorio donde esta instalado Hadoop

- Iniciar el NameNode y DataNode

- `> sbin/start-dfs.sh`

- Deberías poder acceder a la web del NameNode en `http://<ip>:50070`

- Importante: Si no accedes, es posible que el firewall esté activado

- Desactivar firewall: `> service iptables stop`

Instalar Hadoop – Pseudo-Distributed

➤ El servicio (demonio) de hdfs se empezará a ejecutar

➤ top

➤ ps aux | grep dfs

➤ free -m

```
top - 05:11:31 up 10:09, 3 users, load average: 0,29, 0,59, 0,47
Tasks: 180 total, 1 running, 179 sleeping, 0 stopped, 0 zombie
%Cpu(s): 2,0 us, 1,3 sy, 0,0 ni, 96,3 id, 0,0 wa, 0,0 hi, 0,3 si, 0,0 st
KiB Mem : 999920 total, 83988 free, 732924 used, 183008 buff/cache
KiB Swap: 2097148 total, 1680648 free, 416500 used. 59604 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
12124	bigdata	20	0	1521664	157980	5404	S	0,7	15,8	3:52.42	gnome-shell
31632	bigdata	20	0	2817228	125340	4888	S	0,7	12,5	0:09.10	java
31008	bigdata	20	0	2087230	114880	23240	S	0,7	11,3	0:12.97	firefox
1134	root	20	0	262776	45964	10988	S	0,3	4,6	1:39.24	Xorg
31786	bigdata	20	0	2801248	39700	4468	S	0,3	4,0	0:07.15	java
32001	root	20	0	0	0	0	S	0,0	0,0	0:00.20	kworker/0:2
1	root	20	0	193624	2132	560	S	0,0	0,2	0:10.41	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:01.79	kthreadd
3	root	20	0	0	0	0	S	0,0	0,0	0:02.10	ksoftirqd/0
7	root	rt	0	0	0	0	S	0,0	0,0	0:00.00	migration/0
8	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcu_bh
9	root	20	0	0	0	0	S	0,0	0,0	0:04.39	rcu_sched
10	root	rt	0	0	0	0	S	0,0	0,0	0:17.69	watchdog/0
12	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kdevtmpfs

Instalar Hadoop – Pseudo-Distributed

Hadoop

Overview

Datanodes

Datanode Volume Failures

Snapshot

Startup Progress

Utilities

Overview 'localhost:9000' (active)

Started:	Fri Jan 22 18:34:19 CET 2016
Version:	2.7.1, r15ecc87cc4a0228f35af09fc56de536e6ce657a
Compiled:	2015-06-29T06:04Z by jenkins from (detached from 15ecc87)
Cluster ID:	CID-b6da5103-057a-41b4-9fd8-dadf83aabdd4
Block Pool ID:	BP-2111977582-172.16.150.129-1453483932990

Summary

Security is off.

Safemode is off.

1 files and directories, 0 blocks = 1 total filesystem object(s).

Heap Memory used 31.63 MB of 53.39 MB Heap Memory. Max Heap Memory is 966.69 MB.

Non Heap Memory used 30.54 MB of 31.94 MB Committed Non Heap Memory. Max Non Heap Memory is 214 MB.

Configured Capacity:

17.11 GB

Instalar Hadoop – Pseudo-Distributed

➤ Ahora creamos los directorios de usuarios en HDFS

➤ > bin/hdfs dfs -mkdir /user

➤ > bin/hdfs dfs -mkdir /user/root

➤ > bin/hdfs dfs -mkdir /user/bigdata

➤ Para probar

➤ > bin/hdfs dfs -put etc/hadoop input

➤ ¿En qué directorio del HDFS se copian los ficheros?

➤ ¿Qué ocurre si no hubiéramos creado el directorio?

Instalar Hadoop – Pseudo-Distributed

➤ Para probar

- `> bin/hdfs dfs -put etc/* /user/root/prueba`
- `> bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.8.1.jar wordcount /user/root/prueba salidaHDFS1`
- `> bin/hdfs dfs -ls /user/root/salidaHDFS1`

➤ Para parar todos los servicios

- `# sbin/stop-dfs.sh`

Instalar Hadoop – Pseudo-Distributed

➤ Para probar

- `> bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.1.jar grep input output 'dfs[a-z.]+'`
- `> bin/hdfs dfs -cat output/*`

➤ Para parar todos los servicios

- `# sbin/stop-dfs.sh`

Instalar Hadoop – Pseudo-Distributed

- ¿Qué son todos estos ficheros de configuración que hemos editado?



<https://hadoop.apache.org/docs/r2.8.0/>

Instalar Hadoop – Pseudo-Distributed

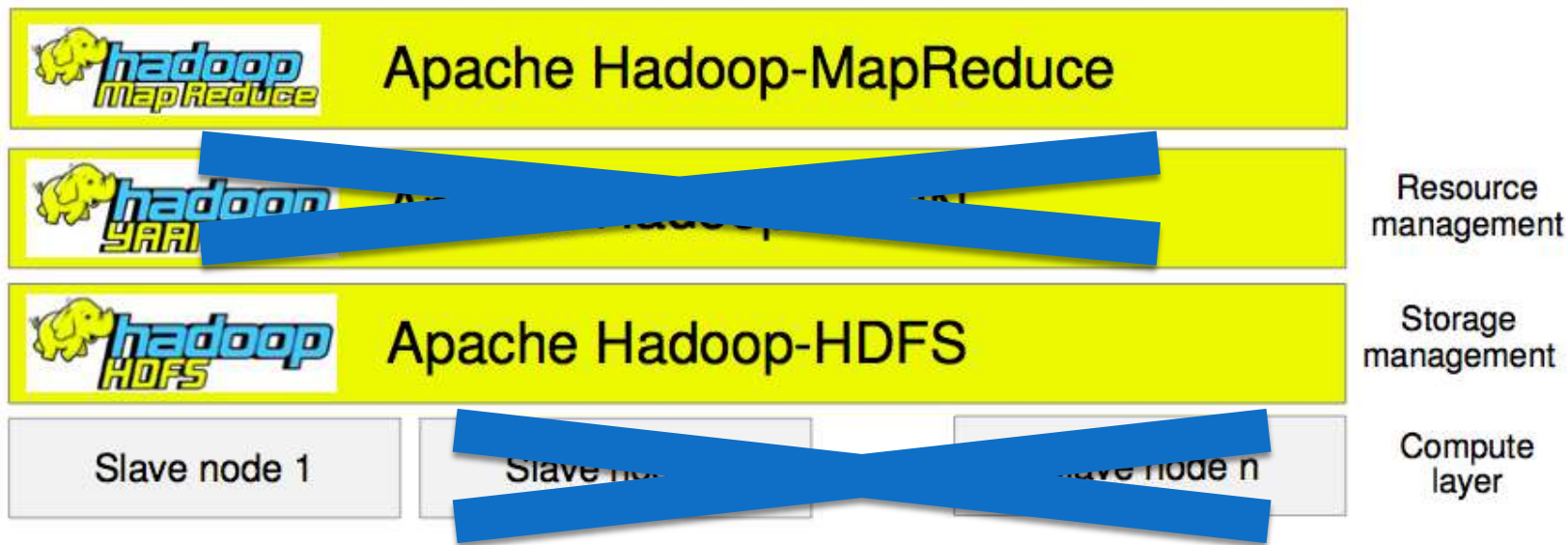
- Un pequeño resumen (<http://www.big-data.tips/hadoop-configuration>)
 - *The Hadoop configuration file **core-site.xml** contains pieces of information about the particular 'Hadoop site' itself. This includes the hostname and port number used for this particular Hadoop instance. Other optional information is the memory allocated for the file system. There can be also memory limits for storing data or more detailed configurations such as the size of read and write buffers.*
- Para investigar todos los parámetros disponibles:
 - <https://hadoop.apache.org/docs/r2.8.0/hadoop-project-dist/hadoop-common/core-default.xml>

Instalar Hadoop – Pseudo-Distributed

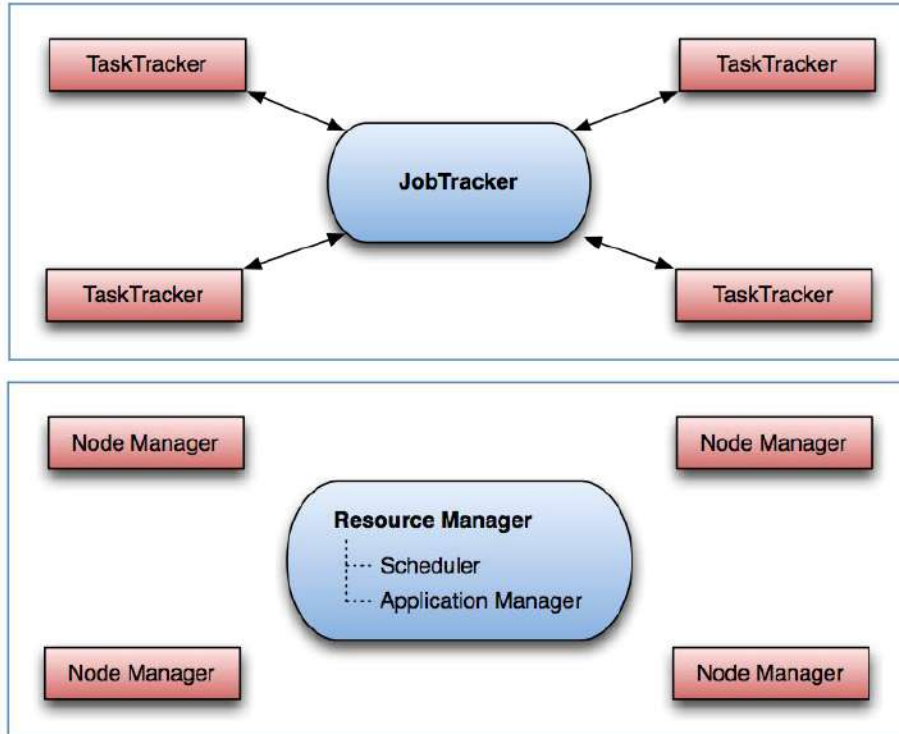
- Un pequeño resumen (<http://www.big-data.tips/hadoop-configuration>)
 - *The Hadoop configuration file **hdfs-site.xml** file contains information about the Hadoop Distributed File System (HDFS) that is part of the Hadoop distribution. It includes the value of ‘replication’ and the path to the ‘namenode’ as well as the paths to ‘datanodes’ based on the local file systems. This is needed in order to tell HDFS a concrete place where data in the Hadoop infrastructure is stored. Below is an example but needs to be configured according to your file system structure depending on the Hadoop infrastructure.*
 - Para investigar todos los parámetros disponibles:
 - <https://hadoop.apache.org/docs/r2.8.0/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml>

Instalación de Hadoop

- Modo *pseudo-distributed* (hasta este punto)



MRv1 vs. MRv2 (Yarn)



- Un JobTracker (master) por cluster
 - Cada esclavo ejecuta un TaskTracker
-
- Single Resource Manager por cluster
 - Cada esclavo ejecuta un Node Manager

Imágenes obtenidas de Cloudera (<http://www.cloudera.com>)

Instalar Hadoop – Pseudo-Distributed con YARN

- La configuración anterior ejecuta tareas MapReduce usando MRv1. Sin embargo, es posible usar MRv2 (o YARN)
- Para ello es necesario ejecutar el servicio ResourceManager y NodeManager
- Partiendo de los pasos realizados antes... tras parar los servicios
- Añade la siguiente configuración a *etc/hadoop/mapred-site.xml*

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

Instalar Hadoop – Pseudo-Distributed con YARN

- Añade la siguiente configuración a `etc/hadoop/yarn-site.xml`

```
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
</configuration>
```

- Iniciar el ResourceManager y NodeManager
 - > `sbin/start-yarn.sh` (para finalizar `sbin/stop-yarn.sh`)
- Deberías poder acceder a la web del ResourceManager en `http://<ip>:8088`

Instalar Hadoop – Pseudo-Distributed

- Un pequeño resumen (<http://www.big-data.tips/hadoop-configuration>)
 - *The Hadoop configuration file **mapred-site.xml** specifies which map-reduce framework is used that is in our example here YARN. Any Hadoop distribution contains a template of the 'mapred-site.xml' file named 'mapred-site.xml.template'. We first copy this template file to the correct name and then add lines as shown below.*
- Para investigar todos los parámetros disponibles:
 - <https://hadoop.apache.org/docs/r2.8.0/hadoop-mapreduce-client/hadoop-mapreduce-client-core/mapred-default.xml>

Instalar Hadoop – Pseudo-Distributed

- Un pequeño resumen (<http://www.big-data.tips/hadoop-configuration>)
 - *The Hadoop configuration file **yarn-site.xml** is used for the Hadoop scheduling system 'Yet Another Resource Negotiator (YARN)'. This component is also an integral part of Hadoop alongside HDFS.*
- Para investigar todos los parámetros disponibles:
 - <https://hadoop.apache.org/docs/r2.8.0/hadoop-yarn/hadoop-yarn-common/yarn-default.xml>

Instalar Hadoop – Pseudo-Distributed con YARN



All Applications

Cluster

About

Nodes

Node Labels

Applications

NEW

NEW SAVING

SUBMITTED

ACCEPTED

RUNNING

FINISHED

FAILED

KILLED

Scheduler

Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes
0	0	0	0	0	0 B	8 GB	0 B	0	8	0	1	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:8192, vCores:32>

Show 20 entries

Search:

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress
----	------	------	------------------	-------	-----------	------------	-------	-------------	----------

No data available in table

Showing 0 to 0 of 0 entries

First Previous

Instalar Hadoop – Pseudo-Distributed

➤ Añadimos más servicios ejecutándose en segundo plano

➤ top

➤ ps aux | grep dfs

➤ free -m

```
top - 05:11:31 up 10:09, 3 users, load average: 0,29, 0,59, 0,47
Tasks: 180 total, 1 running, 179 sleeping, 0 stopped, 0 zombie
%Cpu(s): 2,0 us, 1,3 sy, 0,0 ni, 96,3 id, 0,0 wa, 0,0 hi, 0,3 si, 0,0 st
KiB Mem : 999920 total, 83988 free, 732924 used, 183008 buff/cache
KiB Swap: 2097148 total, 1680648 free, 416500 used. 59604 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
12124	bigdata	20	0	1521664	157980	5404	S	0,7	15,8	3:52.42	gnome-shell
31632	bigdata	20	0	2817228	125340	4888	S	0,7	12,5	0:09.10	java
31008	bigdata	20	0	2087230	114880	23240	S	0,7	11,3	0:12.97	firefox
1134	root	20	0	262776	45964	10988	S	0,3	4,6	1:39.24	Xorg
31786	bigdata	20	0	2801248	39700	4468	S	0,3	4,0	0:07.15	java
32001	root	20	0	0	0	0	S	0,0	0,0	0:00.20	kworker/0:2
1	root	20	0	193624	2132	560	S	0,0	0,2	0:10.41	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:01.79	kthreadd
3	root	20	0	0	0	0	S	0,0	0,0	0:02.10	ksoftirqd/0
7	root	rt	0	0	0	0	S	0,0	0,0	0:00.00	migration/0
8	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcu_bh
9	root	20	0	0	0	0	S	0,0	0,0	0:04.39	rcu_sched
10	root	rt	0	0	0	0	S	0,0	0,0	0:17.69	watchdog/0
12	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kdevtmpfs

Instalar Hadoop – Pseudo-Distributed con YARN

- Ahora podrías volver a lanzar la aplicación MapReduce de prueba de nuevo y ver su evolución en la web del Resource Manager
 - `> bin/hdfs dfs -rm -r output`
 - `> bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.1.jar grep input output 'dfs[a-z.]+'`
 - `> bin/hdfs dfs -cat output/*`
- **O si echas de menos el 'wordcount'...**
 - `> bin/hdfs dfs -rm -r output`
 - `> bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.1.jar wordcount input output`
 - `> bin/hdfs dfs -cat output/*`

Instalar Hadoop – Pseudo-Distributed con YARN

- La aplicación debe aparecer en la web...



All Applications

Cluster

- About
- Nodes
- Node Labels
- Applications
- NEW
- NEW SAVING
- SUBMITTED
- ACCEPTED
- RUNNING
- FINISHED
- FAILED
- KILLED
- Scheduler

Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes
1	0	1	0	7	8 GB	8 GB	0 B	7	8	0	1	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:8192, vCores:8>

Show 20 entries

Search:

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress
application_1453722155914_0001	root	grep-search	MAPREDUCE	default	Mon Jan 25 12:45:48 +0100 2016	N/A	RUNNING	UNDEFINED	

Showing 1 to 1 of 1 entries

First Previous

Instalar Hadoop – Pseudo-Distributed con YARN



Logged in as: dr.who

Application application_1453722155914_0001

Cluster

[About](#)
[Nodes](#)
[Node Labels](#)
[Applications](#)
[NEW](#)
[NEW SAVING](#)
[SUBMITTED](#)
[ACCEPTED](#)
[RUNNING](#)
[FINISHED](#)
[FAILED](#)
[KILLED](#)

[Scheduler](#)

Tools

Kill Application

Application Overview

User: root
Name: grep-search
Application Type: MAPREDUCE
Application Tags:
YarnApplicationState: RUNNING: AM has registered with RM and started running.
FinalStatus Reported by AM: Application has not completed yet.
Started: lun ene 25 12:45:48 +0100 2016
Elapsed: 47sec
Tracking URL: [ApplicationMaster](#)
Diagnostics:

Application Metrics

Total Resource Preempted: <memory:0, vCores:0>
Total Number of Non-AM Containers Preempted: 0
Total Number of AM Containers Preempted: 0
Resource Preempted from Current Attempt: <memory:0, vCores:0>
Number of Non-AM Containers Preempted from Current Attempt: 0
Aggregate Resource Allocation: 280848 MB-seconds, 226 vcore-seconds

Show 20 entries

Search:

Attempt ID	Started	Node	Logs
appattempt_1453722155914_0001_000001	Mon Jan 25 12:45:48 +0100 2016	http://hadoop-master:8042	Logs

Showing 1 to 1 of 1 entries

First Previous 1 Next Last

Instalar Hadoop – Pseudo-Distributed con YARN

Cluster

About

Nodes

Node Labels

Applications

NEW

NEW SAVING

SUBMITTED

ACCEPTED

RUNNING

FINISHED

FAILED

KILLED

Scheduler

Tools

Application Attempt Overview

Application Attempt State: RUNNING

AM Container: container_1453722155914_0001_01_000001

Node: hadoop-master:51320

Tracking URL: ApplicationMaster

Diagnostics Info:

Application Attempt Metrics

Application Attempt Headroom : <memory:0, vCores:0>

Total Allocated Containers: 7

Each table cell represents the number of NodeLocal/RackLocal/OffSwitch containers satisfied by NodeLocal/RackLocal/OffSwitch resource requests.

	Node Local Request	Rack Local Request	Off Switch Request
Num Node Local Containers (satisfied by)	6		
Num Rack Local Containers (satisfied by)	0	0	
Num Off Switch Containers (satisfied by)	0	0	1

Total Outstanding Resource Requests: <memory:23552, vCores:23>

Priority	ResourceName	Capability	NumContainers	RelaxLocality	NodeLabelExpression
20	hadoop-master	<memory:1024, vCores:1>	23	true	
20	*	<memory:1024, vCores:1>	23	true	
20	/default-rack	<memory:1024, vCores:1>	23	true	

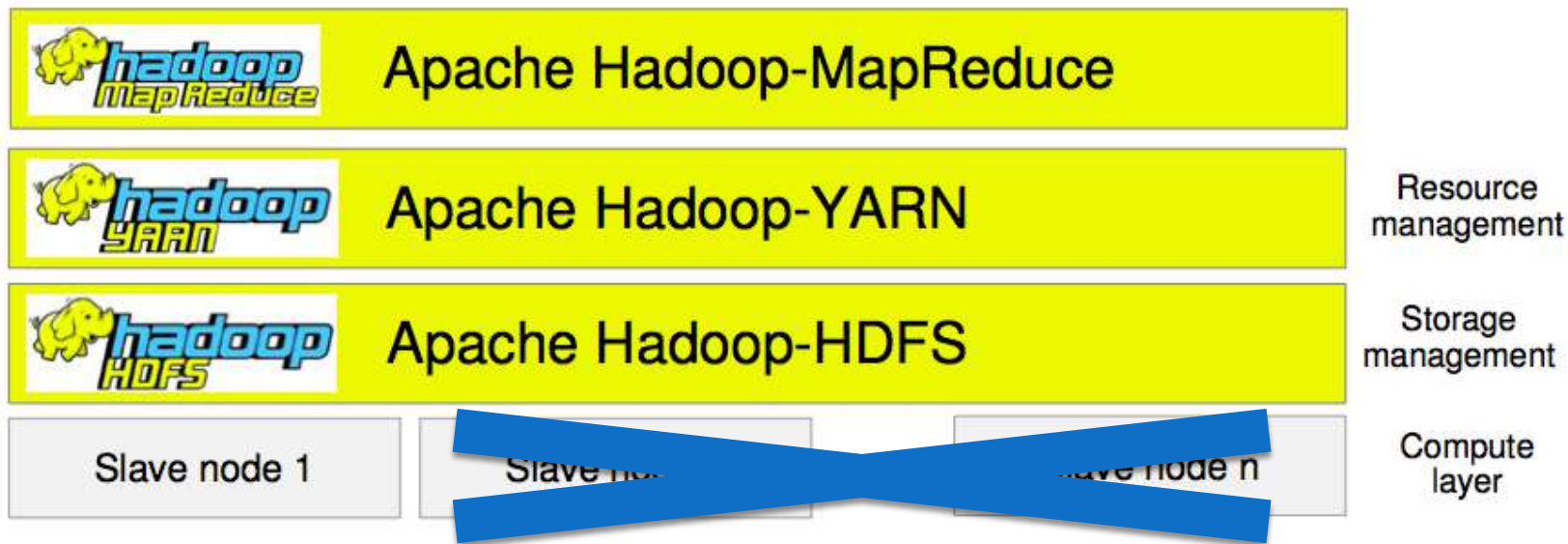
Show 20 entries

Search:

Container ID	Node	Container Exit Status	Logs
container_1453722155914_0001_01_000007	http://hadoop-master:8042	0	Logs
container_1453722155914_0001_01_000006	http://hadoop-master:8042	0	Logs
container_1453722155914_0001_01_000005	http://hadoop-master:8042	0	Logs
container_1453722155914_0001_01_000004	http://hadoop-master:8042	0	Logs
container_1453722155914_0001_01_000003	http://hadoop-master:8042	0	Logs

Instalación de Hadoop

- Modo *pseudo-distributed* (ahora)



Instalación de Hadoop

- Instalación *Fully Distributed*
 - Los demonios de Hadoop se ejecutan en un cluster de máquinas
 - HDFS se emplea para distribuir datos entre todos los nodos
 - A menos que se emplee un cluster pequeño (menos de 10 o 20 nodos), el NameNode y JobTracker deben ejecutarse en nodos dedicados
 - Para pequeños clusters pueden ejecutarse en el mismo nodo

**TO BE
CONTINUED...** ➡